are properly inserted into the IC sockets. Check that there are no pins outside the sockets and that there are no pins bent under the IC´s.

Configure the RAM board with all the address switches OFF. Install the RAM board in the computer and then turn on the power and depress and release the reset switch. The LED should NOT be lit. If it is, turn off the computer and locate the problem before proceeding.

Select an unused address region in your computer and configure the address switches on the RAM board for that region. Plug the board in and turn on the computer power. Do not execute any software which initializes the RAM board. [E.g., the HORIZON DOS will initialize all RAM in the computer. To bypass this, boot up with only 8K of the board (addressed at 2000H) turned ON, and turn on the other 8K after boot load.] Next, examine cells in the address region of the memory board using a monitor program or control panel. Eventually, you will examine a cell which powered up with a parity error. When this occurs, the LED should light. Now turn off the computer power. If the LED did not light, then locate and correct the problem before continuing.

P11. Power up the computer with the RAM-16-A board installed in the motherboard. Use a program such as the standard HORIZON DOS initialization to store into each byte on the board, to clear all incorrect parity bytes. Again, enable the parity detection by executing the two instructions shown in the previous step. (It is normal for the LED to turn on momentarily during this sequence: it should turn off when the arming instruction is executed.) It should not now be possible to turn on the LED by examining cells on the RAM board. If the LED does turn on, then either the memory is making errors or there is a problem with the parity option. Locate and correct the problem before continuing.

The parity option is now assembled and checked out.

CONFIGURATION

This section describes each of the RAM-16-A options which must be configured before use of the RAM-16-A. Note that configuration of the parity logic and bank switching logic is described in later sections.

ADDRESS SELECTION

Address selection for the RAM-16-A is determined by the switches at location 7A. To configure the RAM-16-A for a contiguous 16K byte region of addresses beginning at an 8K boundary, two adjacent switches should be ON (right side depressed) and the other six switches should be OFF (left side depressed), according to the following table:

Address Region	Switches on
0000-3FFF	1 and 2 (1st and 2nd)
2000-5FFF	2 and 3 (2nd and 3rd)
4000-7FFF	3 and 4 (3rd and 4th)
6000-9FFF	4 and 5 (4th and 5th)
8000-BFFF	5 and 6 (5th and 6th)
A000-DFFF	6 and 7 (6th and 7th)
C000-FFFF	7 and 8 (7th and 8th)

Note: Certain pairs of non-adjacent 8K address regions starting on 8K boundaries can be selected if, using the following table, exactly one switch is ON from each column:

Region	Switch	Region	Switch
0000-1FFF	1 (1st)	2000-3FFF	2 (2nd)
4000-5FFF	3 (3rd)	6000-7FFF	4 (4th)
8000-9FFF	5 (5th)	A000-BFFF	6 (6th)
C000-DFFF	7 (7th)	E000-FFFF	8 (8th)

Z80 OR 8080 COMPATIBILITY

The RAM-16-A must be configured to specify whether it is being used with a Z80 or 8080 microprocessor. Connect pin 1 to pin 2 on the DIP header at location 7D if Z80 or Z80A operation is required. Do not connect pins 1 and 2 if 8080 operation is required. (Note that pin 1 should be connected to pin 3 in either case, if bank switching will not be used.)

SIGNAL GROUNDING

It is strongly recommended that bus pins 20, 61, and 70 be connected to ground on the computer motherboard. Some S-100 computers (e.g., the HORIZON) already do this. For each of the three bus pins that are connected to ground, connect the appropriate "G" jumper on the RAM-16-A. The "G" jumpers are located at positions 2E, 5E, and 6E.

PHANTOM MEMORY

The RAM-16-A board can be used with some area of ROM superimposed over the address region of the board. If the PH jumper (near location 4E) is installed, then a memory reference to the board will be inhibited if the backplane signal PHANTOM (motherboard pin 67) is LOW. Do not connect the PH jumper unless you intend to use this feature.

PARITY ERROR ACTION

The action taken when a parity error occurs (when armed) is determined by connecting a jumper wire between "PE" (near location 3E) and one of the following labeled locations:

PINT/ Causes an interrupt request in systems not using vectored interrupts.

NMI/ Causes a non-maskable interrupt request in Z8Ø computers.

VIØ/-VI7/ Connecting to one of these eight locations causes a vectored interrupt request at the corresponding priority level.

Note: The on-board LED lights whenever a parity error is detected, whether armed or not.

PARITY ERROR ARMING

The parity error logic is disarmed at power-on and reset. The parity error logic can be armed or disarmed under program control by loading a control value into the A-register and executing an OUT ØCØH instruction. Bit Ø of the control value should be 1 to arm the parity logic and Ø to disarm it. Configuration of the DIP header at location 7D will determine which bit (of bits 1-7) in the control value, if 1, will cause arming or disarming of the parity logic for the RAM-16-A. (Arming or disarming the parity error logic also resets the parity error flip-flop.)

Parity Select Bit	Header Connection
Bit 1	pin 4 to pin 14
Bit 2	pin 4 to pin 11
Bit 3	pin 4 to pin 9
Bit 4	pin 4 to pin 12
Bit 5	pin 4 to pin 1Ø
Bit 6	pin 4 to pin 8 (standard)
Bit 7	pin 4 to pin 13

Note that the standard HORIZON convention is to use bit 6 for all RAM-16-A boards in the computer. Using this convention, the following two instructions will reset then arm the parity logic for all the RAM boards:

and the following two instructions will reset and disarm the parity logic:

```
      MVI A,4ØH
      OUT ØCØH
```

After power-on, the memory bytes will contain random values (including the parity bit), and not all bytes will have correct parity. Before arming the parity logic, clear all bytes to correct parity by storing into all bytes of RAM. Note that the standard HORIZON DOS initialization performs this function.

USING BANK SWITCHING

At any time, a RAM-16-A may be in one of two "states":

ON In this state the board will respond to memory references made to the addresses specified by the address selection switches.

OFF In this state the board will ignore all memory references from the processor. However, all values on the board will be retained, and refresh cycles will continue.

Thus, it is possible for more than one board to share the same address region in the computer. However, for any particular address region, at most one RAM-16-A should be ON at any given moment.

Configuration of the DIP header at location 7D determines if the board powers up in the ON state (connect pin 6 to pin 7) or in the OFF state (connect pin 5 to pin 6).

The RAM-16-A can be turned on or off under program control by loading a control value into the A-register and executing an OUT ØCØH instuction. Bit Ø of the control value should be Ø to turn the board ON and 1 to turn the board OFF. Configuration of the DIP header at location 7D will determine which bit (of bits 1-7) of the control value, if 1, will cause the board to be set:

Select Bit	Header Connection
Bit 1	pin 3 to pin 14
Bit 2	pin 3 to pin 11
Bit 3	pin 3 to pin 9
Bit 4	pin 3 to pin 12
Bit 5	pin 3 to pin 1Ø
Bit 6	pin 3 to pin 8
Bit 7	pin 3 to pin 13

THEORY OF OPERATION

The RAM-16-A consists of a 4 by 8 (4 by 9 if parity is included) array of 2ØØns 4K dynamic 4Ø27 RAM chips plus additional circuitry which performs the necessary support functions. Refer to the schematic drawings while reading the following theory of operation.

TYPES OF CYCLES

THE RAM-16-A employs three types of cycles in its operation. In each case the cycle is initiated by the CYCLE-START signal which is passed through the tapped delay module to provide the cycle timing.

1. Normal memory cycles are RAS,CAS cycles, which use A5-AØ as row address bits, then A11-A6 as column address bits. If the signal WE is asserted, a store cycle is performed, otherwise a fetch cycle is performed.

2. Deselect cycles are CAS-only cycles, performed during normal memory cycles on all BUT the selected chips. The CAS without a preceding RAS turns off the output drivers so as not to interfere with the data from the selected chips.

3. Refresh cycles are RAS-only cycles, performed by all chips on the board simultaneously. The row address comes from the 74LS393 refresh counter.

CYCLE TIMING

The following is the nominal (disregarding gate delays) timing for a normal memory cycle with respect to CYCLE-START:

Time	Event
Øns	Begin cycle, allow address and control signals to stabilize.
35ns	Begin RAS pulse to selected chips.
6Øns	Switch address multiplexers to column inputs.
11Øns	Begin CAS pulse to all chips.
215ns	Begin CYC-END pulse.
245ns	Terminate RAS
32Øns	Terminate CYC-END, terminate CAS.

CONDITIONS LEADING TO CYCLE S

1. The INSTRUCTION-FETCH flip-flop is set by the leading edge of SM1. For a Z8Ø processor, the instruction fetch presents the tightest access time requirement (even though SM1 precedes SMEMR).

2. For a Z8Ø processor, the FETCH flip-flop is set by the leading edge of SMEMR.

3. For an 8Ø8Ø processor, the FETCH flip-flop is set by the AND of PSYNC, PHI 1, and D7 which is an anticipation of MEMR status.

4. Note that it is possible to have INSTRUCTION-FETCH and FETCH set simultaneously, a condition which is redundant but harmless.

5. The STORE flip-flop is set by the leading edge of PWR unless blocked by SOUT. STORE causes a CYCLE-START and a WE.

6. A control panel deposit is detected as the AND of MWRITE and SMEMR, which causes first a DEP-RQ and then a DEPOSIT-CY. The deposit cycle is delayed if it would otherwise immediately follow a waiting refresh cycle.

7. The RUNNING-REFRESH flip-flop is set by the trailing edge of SM1 and provides refresh cycles when the processor is running. As long as there are 64 instructions executed in any 2ms period, this type of cycle will satisfy the refresh requirements.

8. Whenever 2Øus (approximately) elapses with no cycles of any kind, the retriggerable one-shot RECENT goes false, allowing a WAITING-REFRESH cycle. This type of cycle therefore maintains refresh activity when the reset switch is depressed, during long wait states (control panel or North Star disk controller), or when an 8Ø8Ø is halted. See the next section for more information.

NOTES ON WAITING-REFRESH

1. The WAITING-REFRESH flip-flop is clocked on the trailing edge of the PHI 2 clock, and is true for only one clock period.

2. The refresh cycle is delayed by a DEP-REQ if a control panel deposit cycle is in progress or will start on the next clock cycle. A refresh and a deposit cycle cannot occur on the same or sequential clock cycles.

3. Since some of its input conditions are asynchronous to the PHI 2 clock, the setup and hold times of the WAITING-REFRESH flip-

flop cannot be guaranteed to be met, and there is a small but finite chance that it will go into a momentary indecisive state. For this reason, a network of a resistor, capacitor, Schmitt inverter, and Schmitt gate filter the output to avoid the chance of sending an erroneous pulse down the delay module. Since a Schottky flip-flop is used, the filter slows down the signal by 10-15ns.

4. Whenever the processor resumes computation following a pause, refresh cycles must be inhibited lest one collide with the first memory cycle. The RAM-16-A recognizes three such cases:

   a. at the end of a wait state, the leading edge of PRDY or XRDY sets the WAIT-EXIT flip-flop, inhibiting refresh cycles. If this flip-flop initially comes up true, CLR will reset it.

   b. at the end of an 8080 halt phase (the Z80 maintains SM1 activity while halted) the trailing edge of SHLTA triggers the HLT/RST-EXIT one-shot, inhibiting refresh cycles for a few microseconds.

   c. at the end of a system reset, the trailing edge of POC or PRESET, if so jumpered, (whichever occurs last) triggers the same one-shot.

BOARD SELECT AND CHIP SELECT

The RAM-16-A occupies two 8K regions of a 64K byte address space. Address bits A15, A14, and A13 go to a one-of-eight decoder, the outputs of which go to eight switches. The switches are grouped by fours, with 1, 3, 5, and 7 connected together as BS-CD/ and 2, 4, 6, and 8 connected together as BS-AB/. The most common configuration would be to have two adjacent switches ON and the other six OFF. However, the two switches do not have to be adjacent, as long as one is connected to BS-AB/ and one to BS-CD/.

The board has four 4K regions corresponding to "lines" of chips labeled A, B, C, and D (one hesitates to call them "rows" or "columns" because that terminology is used to designate bit arrays within the chips). For a normal memory cycle, only one line should receive a RAS; other lines do a deselect (CAS-only) cycle. The line which is selected for a memory cycle is determined by combining BS-AB and BS-CD with address bit A12, while the RAS timing comes from the delay module. Note, however, that during a refresh cycle REF-SEL causes all four lines to receive a RAS but not a CAS.

Memory cycles to all four lines on the board are inhibited if OCCLUDE is true (see Bank Switching below), as none of the outputs of the decoder can go low. Similarly, if the PH jumper is installed, all memory cycles are inhibited whenever the backplane signal PHANTOM (pin 67) is low. PHANTOM is used in some systems to superimpose a ROM over areas normally occupied by RAM.

BANK SWITCHING

Provision is made for a system with more than 64K bytes of memory by allowing more than one board to occupy a given address region, as long as only one board responds to any memory reference. This is accomplished by the flip-flop OCCLUDE which, if set, makes the entire board "invisible" to the processor.

The OCCLUDE flip-flop may be set or reset by an OUT instruction to port CØ hex. The port number is hardwired and may only be changed by rearranging the address inputs to the board. Data bit Ø goes to the D input of this flip-flop. The clock to the flip-flop may be gated by any of the other seven data bits (selection made by jumper). Thus, we have a theoretical maximum of seven "banks" of 64K, or 28 boards.

The CLR signal can initialize OCCLUDE either ON or OFF.

PARITY

The PARITY-ARM flip-flop is programmed in the same manner as OCCLUDE: its D input comes from data bit Ø and its clock can be gated via a jumper by any of the other seven data bits. The same pulse that clocks PARITY-ARM on its trailing edge clears the PARITY-ERROR flip-flop.

PARITY-ERROR is set during a memory fetch cycle if a byte with even parity is read and, via a driver, lights the on-board LED. The AND of PARITY-ARM and PARITY-ERROR goes to another driver, the output of which the user may wire to one of the eight vectored interrupts, to PINT, or to NMI.

The CLR signal initializes the PARITY-ERROR and PARITY-ARM flip-flops off.

SPECIAL DMA APPLICATIONS

This section describes modifications for the RAM-16-A which may be necessary for use with such DMA devices as the Cromemco DAZZLER^{TM}. These modifications are only needed if the RAM-16-A is being used with a Z80 or Z80A processor board. (To use the RAM-16-A with a special DMA device in an 8080-based system, it is necessary only to install a resistor as described in step D1 below.)

If the DMA device is simulating an 8080 memory cycle, DO7 must be true during PSYNC. If the device does not drive DO7, a pull-up resistor (say, 1K) to Vcc will suffice. A logical place to add this resistor would be at the DMA controller.

D2. Install resistor R12 (220 ohm, red-red-brn) at location 5E and capacitor C10 (33pF dipped mica) at location 5E. This R-C circuit acts as a delay for the CC-DBSL/ signal.

D3. Add an unused inverter to the circuit by connecting a jumper wire between the location labeled PHLDA (location 7E) to 2C pin 11 and another jumper wire from 2C pin 10 to 2D pin 12. Now cut the trace from 2C pin 12 to 2C pin 10.

Add a second inverter to the circuit by connecting a jumper wire from the location labeled CC-DSBL/ (location 4E) to 7C pins 4 and 5 and another jumper wire from 7C pin 6 to 7D pin 1. Also, cut the trace on the solder side of the PC board between 7D pin 1 and 7C pin 7. This allows memory cycles to start on the condition:

PSYNC AND PHI 1 AND CC-DSBL/

D5. On the header at location 7D, be sure the jumper from pin 1 to pin 2 is installed, as it should be for Z80 operation.

D6. Mark the changes on the schematic drawings.

APPENDIX 1. PULSE SIGNAL DETECTION

Some steps in the checkout procedure will require test equipment capable of distinguishing a signal containing pulses from a DC signal. Any one of the following will suffice.

1. Use of an oscilloscope is best since the shape and frequency of the pulses can also be determined.

2. Use a logic probe that detects pulses.

3. If the RAM-16-A is being assembled for use with a HORIZON, then use a counter on the motherboard to divide the frequencies down to the audio range and then play the result through a hi-fi amplifier. To do this, remove the 74LS161 at location 7D on the motherboard. Then attach the test probe wire to jumper 2D pin 16 (this is the input to the divider). Next, take the output of the divider at jumper 1ØA pin 11 and connect to the AUX input of the audio amplifier. Finally, connect the AUX input ground on the audio amplifier to signal ground on the motherboard. This arrangement will divide high frequency signals by 4Ø96 and thus put the resulting signal in the audible range. Thus a 4 MHz signal will be heard as a tone one octave higher than a 2MHz signal.

4. Construct the "probe" shown in figure 1C on a piece of cardboard or perf-board. This probe converts high frequency signals to DC signals. The voltage of the resulting DC signal will be proportional to the duty factor of the tested wave form.

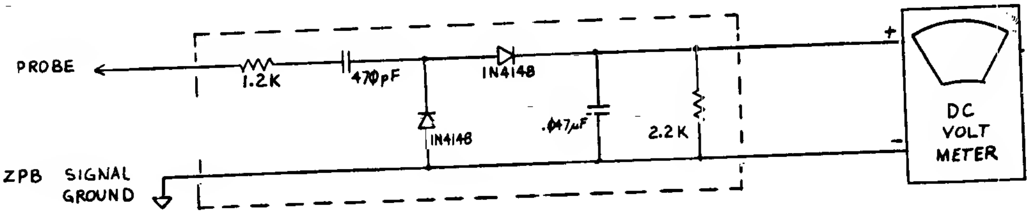


FIGURE 1C: A.C. SIGNAL DETECTOR

APPENDIX 2: MEMORY TEST PROGRAM

```
ØØØØ          *
ØØØØ          *SUBROUTINE TO TEST 4K BLOCK OF RAM
ØØØØ          *  (MUST BE ORIGINED ON 256-BYTE BLOCK BOUNDARY)
ØØØØ          *
ØØØØ          *ENTER WITH BLOCK ADDRESS IN DE (MUST BE 4K BOUNDARY)
ØØØØ          *
ØØØØ          *NOTE THAT ERROR ACTION ROUTINE MUST BE SUPPLIED
ØØØØ          *
ØØØØ ØlEDØØ   TB LXI B,PATTERN  LOAD B,C WITH PATTERN TABLE PTR
ØØØ3 6Ø          MOV H,B        LOAD H WITH MSB OF PTR
ØØØ4 69       PASS MOV L,C      LOAD L WITH LSB OF PTR
ØØØ5          *  BEGIN WRITE FORWARD TO 4K RAM LOOP
ØØØ5 7E       WFL MOV A,M       LOAD A WITH NEXT BYTE OF PATTERN
ØØØ6 12          STAX D         WRITE IT TO RAM UNDER TEST
ØØØ7 2C          INR L          STEP PATTERN PTR
ØØØ8 C2ØEØØ      JNZ WFI        SKIP IF NOT AT END OF PATTERN
ØØØB 21EDØØ      LXI H,PATTERN  RESET PATTERN TABLE PTR
ØØØE 1C       WFI INR E         STEP RAM PTR
ØØØF C2Ø5ØØ      JNZ WFL        LOOP IF NOT AT 256 BOUNDARY
ØØ12 14          INR D          STEP MSB OF RAM PTR
ØØ13 7A          MOV A,D        TEST MSB OF PTR
ØØ14 E6ØF        ANI 17Q        DONE FILLING 4K?
ØØ16 C2Ø5ØØ      JNZ WFL        LOOP IF NOT AT 4K BOUNDARY
ØØ19 7A          MOV A,D        RESET 4K RAM PTR TO BEGINNING
ØØ1A D61Ø        SUI 2ØQ        "     "     "     "     "
ØØ1C 57          MOV D,A        "     "     "     "     "
ØØ1D 69          MOV L,C        RESET PATTERN PTR
ØØ1E          *  BEGIN READ FORWARD AND COMPARE LOOP
ØØ1E 46       RFL MOV B,M       GET PATTERN VALUE
ØØ1F 1A          LDAX D         GET BYTE FROM RAM
ØØ2Ø B8          CMP B          CMP BYTE WITH PATTERN
ØØ21 C467ØØ      CNZ ERROR
ØØ24 2C          INR L          STEP PATTERN PTR
ØØ25 C22BØØ      JNZ RFI        SKIP IF NOT AT END OF PATTERN
ØØ28 21EDØØ      LXI H,PATTERN  RESET PATTERN TABLE PTR
ØØ2B 1C       RFI INR E         STEP RAM PTR
ØØ2C C21EØØ      JNZ RFL        LOOP IF NOT AT 256 BOUNARY
ØØ2F 14          INR D          STEP MSB OF PTR
ØØ3Ø 7A          MOV A,D        TEST MSB OF PTR
ØØ31 E6ØF        ANI 17Q        DONE WITH 4K REGION?
ØØ33 C21EØØ      JNZ RFL        LOOP IF NOT AT 4K BOUNDARY
ØØ36          *  END OF READ LOOP
ØØ36          *
```

```
ØØ36                *
ØØ36                *WE HAVE COMPLETED A READ AND WRITE PASS
ØØ36                *NOW TEST BACKWARDS TO CATCH ADDRESSING ERRORS
ØØ36
ØØ36 69                 MOV L,C         DECR RAM PTR
ØØ37 1B             WBL DCX D           BEGIN BACKWARDS WRITE
ØØ38 7E                 MOV A,M         LOAD NEXT BYTE OF PATTERN
ØØ39 12                 STAX D          WRITE IT TO RAM
ØØ3A 2C                 INR L           STEP PATTERN PTR
ØØ3B C241ØØ             JNZ WBI         SKIP IF NOT AT END OF TABLE
ØØ3E 21EDØØ             LXI H,PATTERN   RESET PATTERN PTR
ØØ41 7A             WBI MOV A,D         GET MSB OF PTR
ØØ42 E6ØF               ANI 17Q         END OF 4K BOUNDARY?
ØØ44 B3                 ORA E           "     "     "
ØØ45 C237ØØ             JNZ WBL         LOOP IF NOT AT BLOCK BEGINNING
ØØ48 7A                 MOV A,D         RESET PTR TO END OF BLOCK
ØØ49 C61Ø               ADI 2ØQ         "     "     "     "
ØØ4B 57                 MOV D,A         "     "     "     "
ØØ4C 69                 MOV L,C         RESTORE TABLE PTR
ØØ4D 1B             RBL DCX D           READ PASS (BACKWARDS)
ØØ4E 46                 MOV B,M         PATTERN
ØØ4F 1A                 LDAX D          DATA
ØØ5Ø B8                 CMP B           COMPARE
ØØ51 C467ØØ             CNZ ERROR
ØØ54 2C                 INR L           STEP TABLE PTR
ØØ55 C25BØØ             JNZ RBI         SKIP IF NOT AT END OF PATTERN
ØØ58 21EDØØ             LXI H,PATTERN   RESET PATTERN TABLE PTR
ØØ5B 7A             RBI MOV A,D         STEP RAM PTR
ØØ5C E6ØF               ANI 17Q         BEGIN TEST
ØØ5E B3                 ORA E           CONTINUE TEST
ØØ5F C24DØØ             JNZ RBL         LOOP IF NOT AT END OF 4K
ØØ62 ØC                 INR C           CHANGE PATTERN
ØØ63 C2Ø4ØØ             JNZ PASS
ØØ66 C9                 RET             ALL DONE WITH TEST
ØØ67                *
```

```
ØØ67
ØØ67            *ERROR ROUTINE (NOT SUPPLIED)
ØØ67            * ON ENTRY, REGISTERS CONTAIN:
ØØ67            *
ØØ67            *  A  VALUE FOUND IN RAM
ØØ67            *  B  VALUE SHOULD HAVE BEEN FOUND
ØØ67            *  C  PASS NUMBER
ØØ67            * DE  BAD BYTE ADDRESS
ØØ67            * HL  ADDRESS OF PATTERN
ØØ67            ERROR DS 2Ø6Q         PUT ERROR SUBROUTINE HERE
ØØED
ØØED            *  ADD YOUR ERROR ROUTINE HERE
ØØED
ØØED            *
ØØED            *PATTERN TABLE
ØØED
ØØED ØØ         PATTERN DB Ø
ØØEE Ø1            DB 1
ØØEF Ø2            DB 2
ØØFØ Ø4            DB 4
ØØF1 Ø8            DB 1ØQ
ØØF2 1Ø            DB 2ØQ
ØØF3 2Ø            DB 4ØQ
ØØF4 4Ø            DB 1ØØQ
ØØF5 8Ø            DB 2ØØQ
ØØF6 AA            DB 252Q
ØØF7 7F            DB 177Q
ØØF8 BF            DB 277Q
ØØF9 DF            DB 337Q
ØØFA EF            DB 357Q
ØØFB F7            DB 367Q
ØØFC FB            DB 373Q
ØØFD FD            DB 375Q
ØØFE FE            DB 376Q
ØØFF FF            DB 377Q        MUST BE END OF 256-BYTE BLOCK
Ø1ØØ            *
```

'PENDIX 3: ORGANIZATION OF RAM CHIP ARRAY

'he organization of RAM chips on the PC board layout is as ollows:

The row 1ØA-17A (9A-17A if the parity option is included) responds to the 4K area of addresses within the 16K region selected by the address select switches with first hex digit of 3, 7, B, or F. For example, if the RAM board select switches "2" and "3" are ON, then this row responds to addresses in the range of 3ØØØ-3FFF hex.

The row 1ØB-17B responds to the 4K area of the 16K address region with first hex digit of 2, 6, A, or E.

The row 1ØC-17C responds to the 4K area of the 16K address region with first hex digit of 1, 5, 9, or D.

The bottom row, 1ØD-17D responds to the 4K area of the 16K address region with first hex digit of Ø, 4, 8, or C.

The columns of RAM chips each correspond to a different bit of each addressed byte:

The column 9A-9D contains the parity bit (if the parity option is included).

The column 1ØA-1ØD contains bit 5 (the 2Ø hex bit).

The column 11A-11D contains bit 4 (the 1Ø hex bit).

The column 12A-12D contains bit 6 (the 4Ø hex bit).

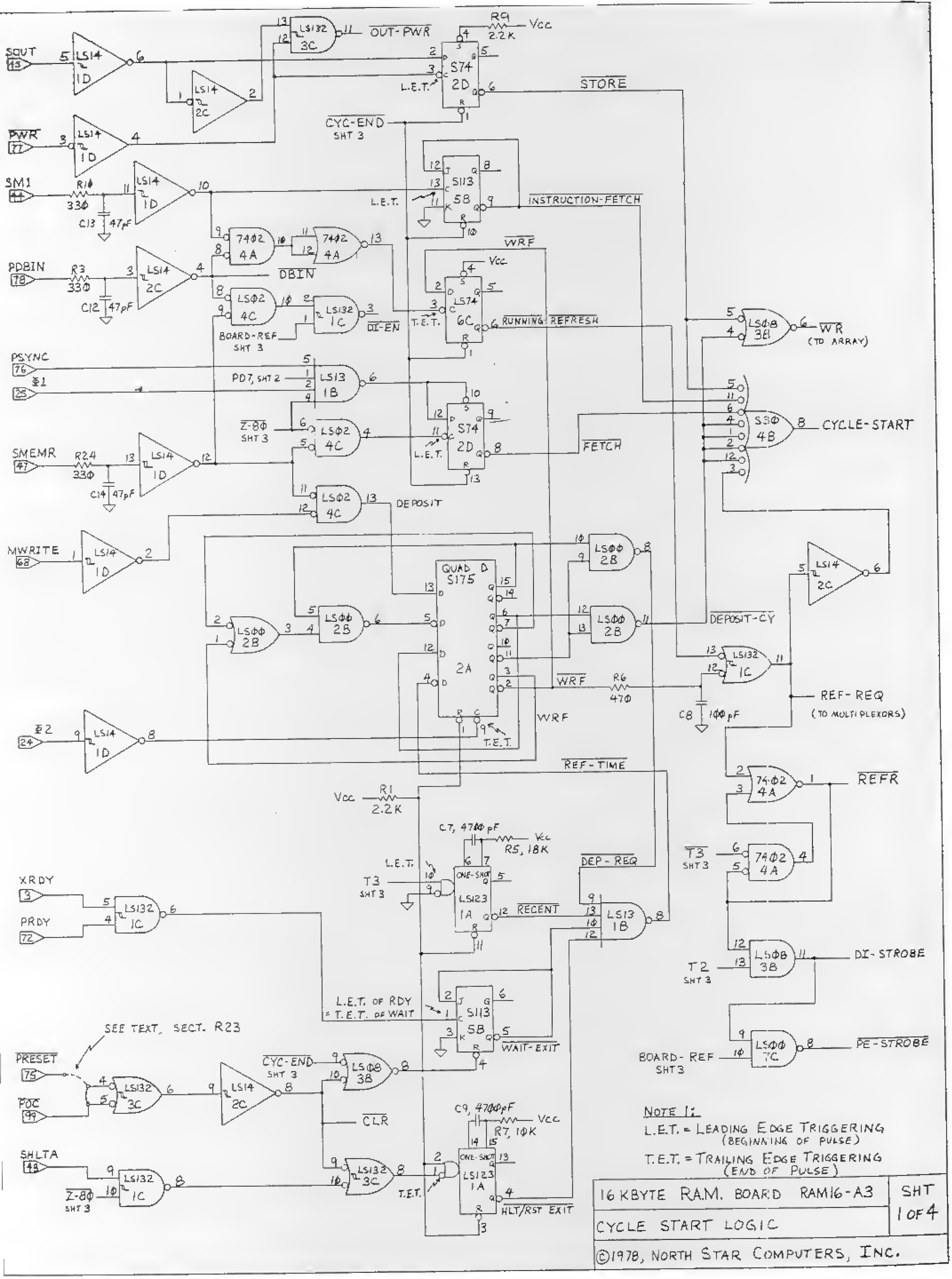
The column 13A-13D contains bit 1 (the Ø2 hex bit).

The column 14A-14D contains bit Ø (the Ø1 hex bit).

The column 15A-15D contains bit 2 (the Ø4 hex bit).

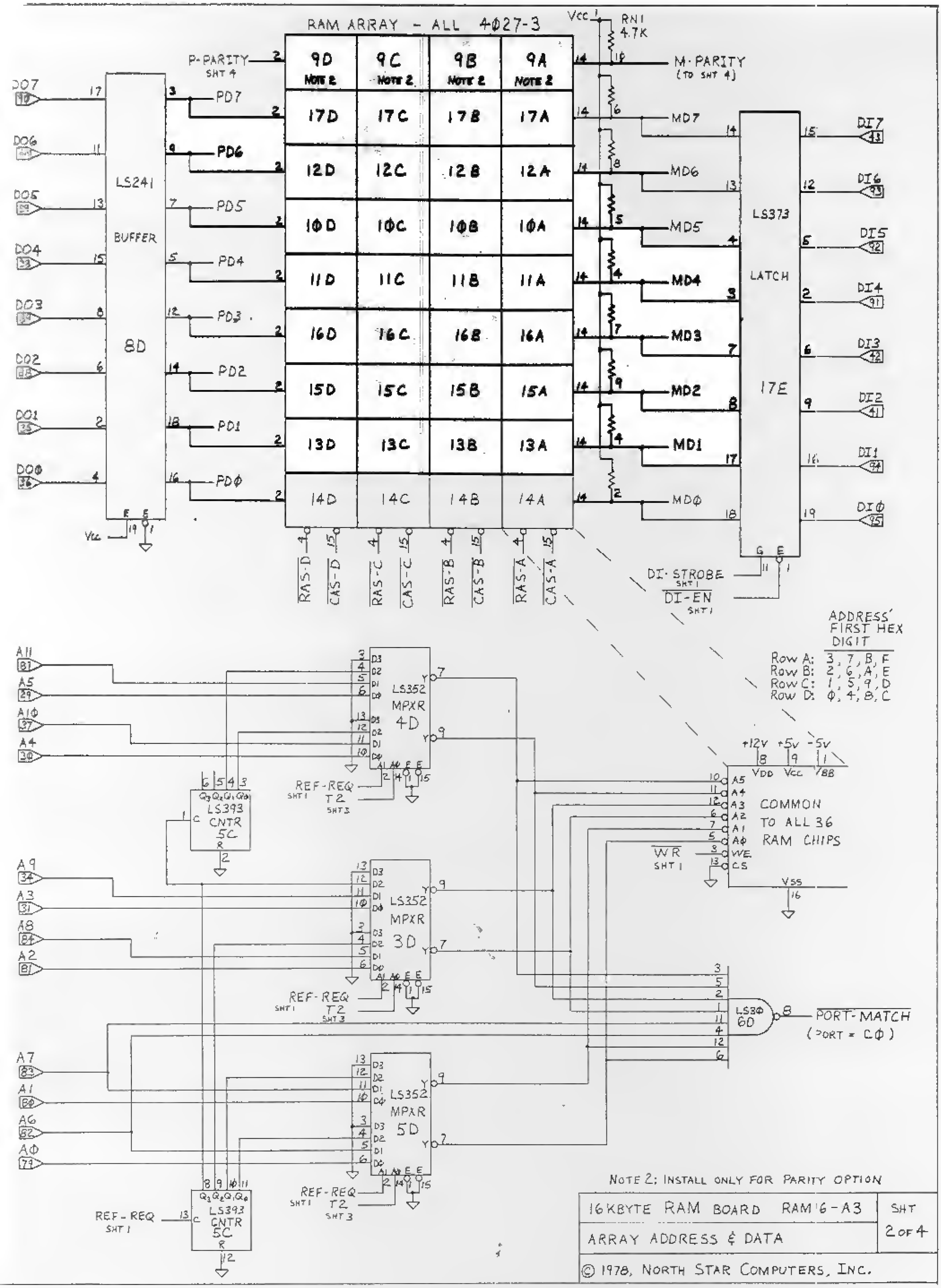
The column 16A-16D contains bit 3 (the Ø8 hex bit).

The column 17A-17D contains bit 7 (the 8Ø hex bit).



NOTE 1:
L.E.T. = LEADING EDGE TRIGGERING (BEGINNING OF PULSE)
T.E.T. = TRAILING EDGE TRIGGERING (END OF PULSE)

16 KBYTE R.A.M. BOARD RAM16-A3 SHT 1 OF 4
CYCLE START LOGIC
©1978, NORTH STAR COMPUTERS, INC.



ADDRESS FIRST HEX DIGIT

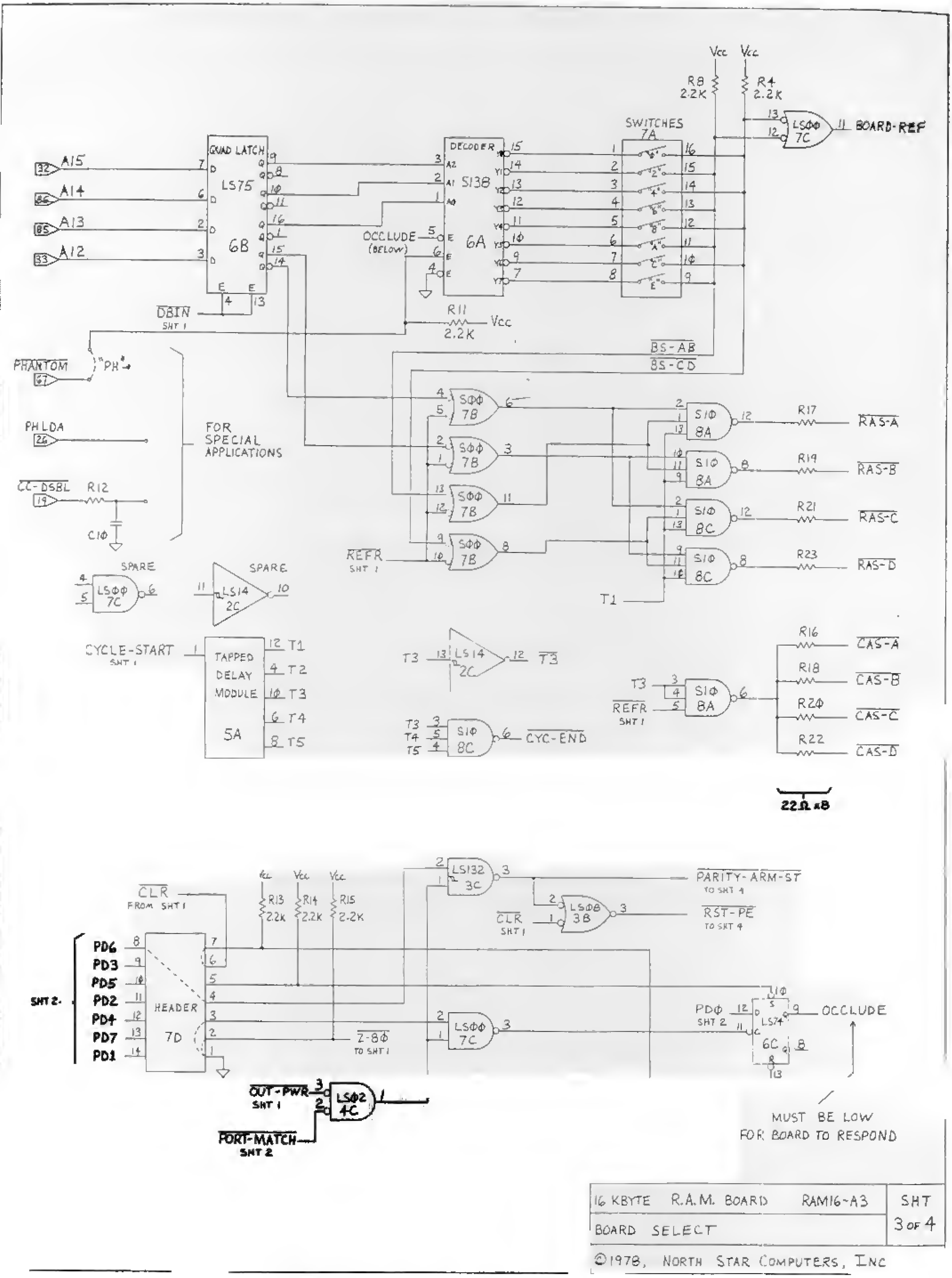
Row A:	3, 7, B, F
Row B:	2, 6, A, E
Row C:	1, 5, 9, D
Row D:	Ø, 4, 8, C

NOTE 2: INSTALL ONLY FOR PARITY OPTION

16KBYTE RAM BOARD RAM16-A3

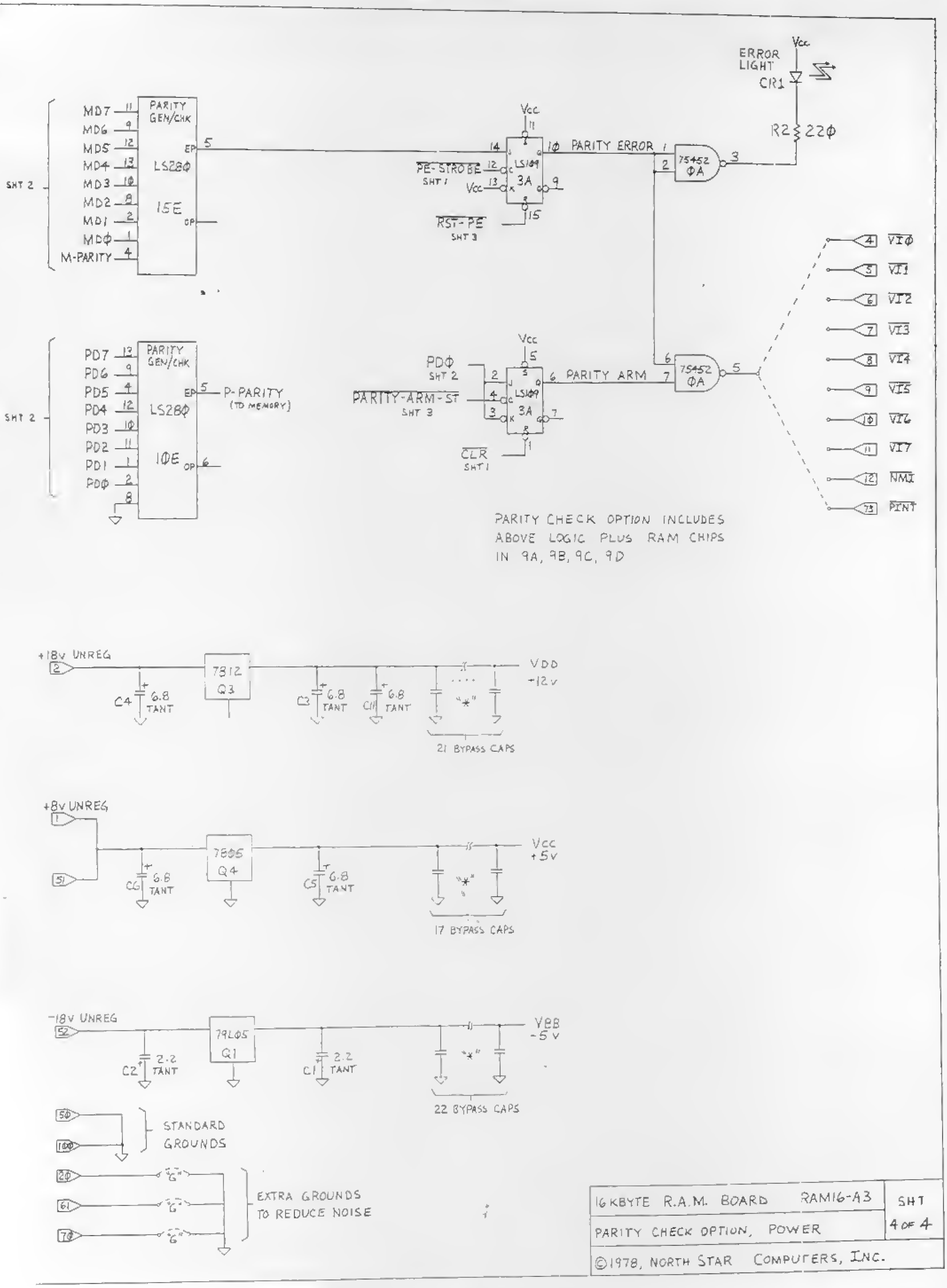
ARRAY ADDRESS & DATA

SHT 2 OF 4



16 KBYTE R.A.M. BOARD RAM16-A3	SHT
BOARD SELECT	3 OF 4

©1978, NORTH STAR COMPUTERS, INC



16KBYTE R.A.M. BOARD RAM16-A3

SHT 4 OF 4

PARITY CHECK OPTION, POWER

©1978, NORTH STAR COMPUTERS, INC.